DII.3200.Sol251.DTK.3201.VDD-1

Defense Information Infrastructure (DII)

Common Operating Environment (COE)

Version 3.2.0.0

Version Description Document for the Developer's Toolkit

Version 3.2.0.1

(Solaris 2.5.1)

September 26, 1997

Prepared for:

Defense Information Systems Agency

Prepared by:

Inter-National Research Institute (INRI)
12200 Sunrise Valley Drive, Suite 300
Reston, Virginia 20191

Table of Contents

This page intentionally left blank.

NOTE: `Courier` font is used to indicate entries to be typed at the keyboard, operating system commands, file and directory names, and screen text. For example:

The file is located in the `DII_DEV` directory.

1. System Overview

Defense Information Infrastructure (DII) Common Operating Environment (COE) developer's tools were developed to aid the developer in the creation and ultimate installation of DII COE segments. The tools make software integration a largely automated process, thus significantly reducing development time while automatically allowing detection of potential integration and runtime problem areas.

By default, developer's tools are located underneath the `DII_DEV` directory and are distributed as part of the Developer's Toolkit. The actual location of the Developer's Toolkit may vary from system to system because it is installed by means of a `tar` command. For example, if the toolkit is tarred to `/h`, the path would be `/h/DII_DEV`.

This delivery consists of the following items:

- 11 developer tools (CalcSpace, CanInstall, ConfigDef, ConvertSeg, MakeAttribs, MakeInstall, TestInstall, TestRemove, TimeStamp, VerifySeg, and VerUpdate)
- `include` files, located in the `include` directory
- public Application Programmer Interface (API) libraries, located in the `libs` directory
- a data file called `TapeSizes`, located in the `data` directory, which contains a list of known tape devices
- an environment setup script called `MakeTOOLSEnv`, located in the `Scripts` directory
- public API examples, located in the `examples` directory; and (10) sample segments, located in the `SampleSegments` directory.

The DII COE Version 3.2.0.0 Developer's Toolkit Version 3.2.0.1 encompasses the following changes:

- The tool ConfigDef has been added to support distributions. Distributions simplify the installation procedure by allowing the user to install logically grouped segments known as bundles or configurations.

- Several tools and APIs have been modified to support Helper Functions. Helper Functions allow the DII COE tools to be extensible to accommodate new technologies. For information on how to create and use helper functions, refer to the *DII COE Version 3.2.0.0 Programmer's Manual for Helper Function*.

- The COE_Is_Permitted API is now supported to allow an application to determine what permissions a particular user has for running the current application.

The following developer's tools have been enhanced to support Helper Functions:

- CanInstall
- MakeInstall
- TestInstall
- TestRemove

.

- VerifySeg.

The following APIs, as described in the *DII COE Version 3.2.0.0 Programmer's Manual for Helper Function*, have been added to support Helper Functions:

- COEHFWarningLog
- COEHFErrorLog
- COEHFDebugLog
- COEHFVerboseLog
- COEHFInstallerLog.

Developer's Tools

The developer's tools can be run from the command line, and some can be run from other code using published APIs. The following tools are included in the DII COE Version 3.2.0.0 Developer's Toolkit Version 3.2.0.1 delivery. The *DII COE Version 3.2.0.0 Programmer's Manual for Kernel Version 3.2.0.0 Patch 1 and Developer's Toolkit Version 3.2.0.1 (Solaris 2.5.1)* describes these tools and their functionality in more detail.

- **CalcSpace** Version 1.0.0.6—Computes the space required for the segment specified and updates the hardware descriptor.
- **CanInstall** Version 1.0.0.9—Tests a segment to see if it can be installed, which means that all required segments must already be on the disk, and the disk cannot have any conflicting segments.
- **ConfigDef** Version 1.0.0.0—Creates a distribution definition from a list of segments.
- **ConvertSeg** Version 1.0.0.9—Examines a segment's segment descriptors and converts them to the *DII COE Integration and Runtime Specification* segment format. Refer to the

DII COE Integration and Runtime Specification Version 2.0 for more information about the DII COE segment format.

- **MakeAttribs** Version 1.0.0.9—Creates the descriptor file `FileAttribs`, which recursively traverses every subdirectory beneath the segment home directory and creates a file with lines in the proper format.

- **MakeInstall** Version 1.0.1.8—Writes one or more segments to an installation medium or packages the segments for distribution over the network.

- **TestInstall** Version 1.0.0.11—Installs a segment (temporarily) that already resides on disk.

- **TestRemove** Version 1.0.0.10—Removes a segment that was installed by TestInstall.

- **TimeStamp** Version 1.0.0.8—Puts the current time and date into the `VERSION` descriptor of the specified segment.

- **VerifySeg** Version 1.0.0.11—Validates that a segment conforms to the rules for defining a segment.

- **VerUpdate** Version 1.0.1.7—Increments the segment version number and updates the date and time in the `VERSION` descriptor of the specified segment.

`MakeTOOLSEnv` Setup Script

The `MakeTOOLSEnv` setup script is located in the `DII_DEV/Scripts` directory. The script defines the environment variables required for developer's tools processing.

`TapeSizes` Data File

The `TapeSizes` data file is located in the `DII_DEV/data` directory. The data file contains a list of known tape devices. This is used as a convenience feature with the MakeInstall tool.

> **NOTE**: The environment variable `TOOLS_DATA` must be set and pointing to `DII_DEV/data` to allow MakeInstall to access this file.

`examples` Directory

The `examples` directory contains a list of example C programming files that show developers how to use the public APIs. The required public `include` files are also shown. Examples are located in the `DII_DEV/examples` directory.

`include` Directory
Public `include` files are used to compile with the public APIs. Public `include` files are located in the `DII_DEV/include` directory.

`libs` Directory
Public API libraries are located in the `DII_DEV/libs` directory.

2. Referenced Documents

The following documents are referenced in this Version Description Document (VDD):

- DII.3200.HPSOL.Helper.PM-1, *Defense Information Infrastructure (DII) Common Operating Environment (COE) Version 3.2.0.0 Programmer's Manual for Helper Function (HP-UX 10.20/Solaris 2.5.1)*, September 26, 1997

- DII.3200.Sol251.PM-1, *Defense Information Infrastructure (DII) Common Operating Environment (COE) Version 3.2.0.0 Programmer's Manual for Kernel Version 3.2.0.0 Patch 1 and Developer's Toolkit Version 3.2.0.1 (Solaris 2.5.1)*, September 26, 1997

- DII COE I&RTS:Rev 2.0, *Defense Information Infrastructure (DII) Common Operating Environment (COE) Integration and Runtime Specification Version 2.0*, October 23, 1995

- DII COE I&RTS:Rev 3.0, *Defense Information Infrastructure (DII) Common Operating Environment (COE) Integration and Runtime Specification Version 3.0*, July 1997

- DII.3200.Sol251.DTK.3201.IG-1, *Defense Information Infrastructure (DII) Common Operating Environment (COE) Version 3.2.0.0 Developer's Toolkit Version 3.2.0.1 Installation Guide (Solaris 2.5.1)*, September 26, 1997

- DII.3200.Sol251.RG-1, *Defense Information Infrastructure (DII) Common Operating Environment (COE) Version 3.2.0.0 Application Programmer Interface (API) Reference Guide (Solaris 2.5.1)*, July 25, 1997.

3. Version Description

3.1 Inventory of Materials Released

- Magnetic media: Two 8mm tapes consisting of a relative tar of the Developer's Toolkit, Version 3.2.0.1 (Solaris 2.5.1)

- Soft-copy Documentation: One 3.5" floppy diskette (WordPerfect 6.0 for Windows format) containing the *Defense Information Infrastructure (DII) Common Operating Environment (COE) Version 3.2.0.0 Version Description Document for the Developer's Toolkit Version 3.2.0.1 (Solaris 2.5.1)*, Document Control No. DII.3200.Sol251.DTK.3201.VDD-1, September 26, 1997

- Hard-copy Documentation: Two hard copies of the aforementioned VDD.

3.2 Software Changes

This delivery contains enhancements for distributions and Helper Functions, as specified in the *Defense Information Infrastructure (DII) Common Operating Environment (COE) Integration and Runtime Specification Version 3.0*. Refer to the following table for a list of other software changes for the DII COE Version 3.2.0.0 Developer's Toolkit Version 3.2.0.1 (Solaris 2.5.1). The software changes are listed in order of INRI Number.

DII COE Version 3.2.0.0 Developer's Toolkit Version 3.2.0.1 (Solaris 2.5.1) Software Changes
Total Software Changes: [2]
INRI Number: RTO00002015 **Agency Number**: D70415 **Short Description**: COE_Is_Permitted API **Long Description**: Currently, when an application is launched, that application cannot determine the permissions that the current user has for that particular application. This API will allow an application to query to see what permissions a particular user has for running the current application.
INRI Number: RTO00001646 **Agency Number**: D70368 **Short Description**: Multiple account groups cannot be specified. **Long Description**: Page 5-94 of the I&RTS that the user may have "one or more $SEGMENTS lines" in the SegName file, which would allow the segment to belong to more than one account group. It appears, however, that only the first $SEGMENT is recognized when a segment is loaded.

4. Installation Instructions

Refer to the *DII COE Developer's Toolkit Installation Guide* for instructions on installing the developer's toolkit.

5. Known Problems and Errors

Refer to the following table for a list of known problems and errors for the DII COE Version 3.2.0.0 Developer's Toolkit Version 3.2.0.1 (Solaris 2.5.1). The known problems and errors are listed in order of INRI Number.

DII COE Version 3.2.0.0 Developer's Toolkit Version 3.2.0.1 (Solaris 2.5.1) Known Problems and Errors
Total Known Problems and Errors: [7]
INRI Number: RTO00001708 **Agency Number**: None. **Short Description**: ConfigDef allows bundling of Child segments without a Parent but should generate an error that an Aggregate is indivisible. **Long Description**: A Child segment can be successfully included in a ConfigDef statement, but no indication is made that the Parent segment must also be included. Once MakeInstall is run, an error is correctly generated. However, if MakeInstall is performed using all of the segments of an Aggregate, the bundle shows up in the Installer with nothing listed under the bundle, and selection of the bundle does not result in the INSTALL button being made available. ConfigDef should generate an error indicating that the Aggregate is indivisible.
INRI Number: RTO00001710 **Agency Number**: None. **Short Description**: ConfigDef is generating incorrect errors when necessary 'options' are not listed. **Long Description**: In several instances, the ConfigDef tool has generated Error notices stating that various options are "invalid" due to the fact that either the '-nb' or '-nc' option was not specified. COMMAND/ERROR: C: ConfigDef -p . -o patches -r "Test20 and 4 patches" -rf Test20/SegDescrip/ReleaseNotes -v Test20/SegDescrip/ReleaseNotes -b E: Unable to build distribution "-r" is an invalid option" C: ConfigDef -p ./HP -o restest.hanger -s INDIA1 -s INDIA2 -s INDIA3 -s INDIA4 -s INDIA5 -s INDIA6 -s INDIA7 E: Unable to build distribution "-s" is an invalid option. ConfigDef should provide the proper error notice that necessary 'options' have not been provided. Generation of these Error notices is not correct because the options work successfully when the -nb command is provided in the command.

DII COE Version 3.2.0.0 Developer's Toolkit Version 3.2.0.1 (Solaris 2.5.1) Known Problems and Errors
INRI Number: RTO00001721 **Agency Number**: None. **Short Description**: Using ConfigDef with the '-vl' option is generating a COEGetCpp error. **Long Description**: ConfigDef generates the error "(E) COEGetCppPath: Error, can't find cpp dir under : /SegDescrip" but still creates a DistribFile when this command with a "-yl" option is used: ConfigDef -vl -o DistribFile -nb TempBundle -r "Test Comments2" -s ECHO::2.0.0.1 The DistribFile was created successfully per the output. If the "-vl" option is not used, the execution contains no error.
INRI Number: RTO00001725 **Agency Number**: None. **Short Description**: Bundle creation is not checking for required/conflicting segments. **Long Description**: The creation of a bundle is successful even if an Aggregate Parent, Aggregate Child, or required segment is not included in the bundle, or if conflicts exist among the segments included in the list. MakeInstall appropriately fails in both cases, but ConfigDef should not have created the bundle. The user should receive an error during the creation of the bundle.
INRI Number: RTO00001958 **Agency Number**: None. **Short Description**: Segments successfully install via TestInstall when the cleanup database is missing. **Long Description**: A segment will successfully install via TestInstall when the hf_cln.idx file is missing from /h/DII_DEV/data. This problem results in the cleanup function not being executed even though the segment is successfully installed. Segments should fail to install if the cleanup database is missing.
INRI Number: RTO00001960 **Agency Number**: None. **Short Description**: MakeInstall does not use the SEG_DIRS variable when used with multiple distribution files. **Long Description**: MakeInstall does not support use of the SEG_DIRS variable to locate distribution files that reside in a directory other than the current working directory. When more than one distribution file is specified on the command line (e.g., MakeInstall -di dist1 dist2 dist3), MakeInstall generates an error when one of the distribution files (e.g., dist2) is in a directory other than the current working directory, even though the directory is pointed to by the SEG_DIRS variable. Workaround: Place all distribution files in the current working directory prior to executing the command.

DII COE Version 3.2.0.0 Developer's Toolkit Version 3.2.0.1 (Solaris 2.5.1) Known Problems and Errors
INRI Number: RTO30000178
Agency Number: None.
Short Description: COE_Is_Permitted cannot locate the Profile directory for profiles that contain spaces.
Long Description: Having a profile name with a space in it is valid, but when the profile directory is created the space is replaced with an underscore (_). The COE_Is_Permitted inspects the LastProfiles_0 file for the current profile list. The LastProfiles_0 file does not have an underscore in the profile name. The COE_Is_Permitted assumes that the profile name and the directory match. Since they do not, the COE_Is_Permitted API cannot find the associated profile directory. Workaround: COE_Is_Permitted will locate the Profile directory if the profile name is one word or if the name is created originally with an underscore vs. a space in it.

6. Release Notes

General

1. The user must log in as `root` to install the Developer's Toolkit tape. By doing this, the user will ensure that the tools have the correct permissions and ownerships to simplify use.

2. The tools do not support the `?` command line parameter for help. Help can be displayed for all tools by using the `-h` or `-H` command line parameters.

3. If the add, append, or delete keywords are used in the community `SegDescrip` entry, the brackets "`{}`" that mark the beginning and end of the text must be the first characters on a new line.

4. The `DII_DEV/Scripts/MakeTOOLSEnv` script must be run to set the `TOOLS_HOME`, `MACHINE`, `MACHINE_OS`, and `MACHINE_CPU` environment variables or these variables must be set by the developer.

Helper Functions

1. Users with new descriptors that have been approved by the DII COE Chief Engineer should ensure that Defense Information Systems Agency (DISA) enters the descriptors into the set of Helper Function databases. To enter this information, the Helper Function developer must provide DISA with the descriptor name and the name of the executable (helper or cleanup) to invoke for each of the corresponding tools. These databases must be delivered from DISA to the developers.

2. Segments that have no unknown descriptors in their `SegInfo` file will not have any new functionality, as the Helper Functions will not be invoked.

3. Helper Function executables for the runtime tools must reside in the `/h/COE/bin` directory.

4. Helper Function executables for the developer's tools must reside in the `/h/DII_DEV/bin` or `/h/COE/bin` directory.

5. When executing the tools using the Helper Function capabilities, the `MACHINE_CPU` environment variable must be set.

6. When running a tool that invokes a Helper Function, the `TOOLS_HOME` environment variable is referenced for the location of the Helper Function `.idx` files (`hf_run.idx`, `hf_dev.idx`, `hf_cln.idx`). If the `TOOLS_HOME` environment variable does not contain the path to the `.idx` files, the tool will default to `/h/COE/data` as the location of the files. The developer must ensure that the `.idx` files are in the directory specified by `TOOLS_HOME` or `/h/COE/data`.

7. The helper function tools use a checksum to validate the hf_dev.idx and hf_cln.idx database files. Therefore, any helper database files created with the integration release of Helper Function Utility Version 1.0.0.0 will not work with the official release of Helper Function Utility Version 1.0.0.1. Developers who have been using Helper Function Utility 1.0.0.0 will have to upgrade to the official release of Helper Function Utility Version 1.0.0.1 in order to use the helper function facility.

ConfigDef Tool

1. ConfigDef does not support the `-vl` command line option.

2. The `SEG_DIRS` environment variable must be set if the `-p` parameter is not used.

ConvertSeg Tool

1. Data descriptors are converted using ConvertSeg, but the developer is responsible for moving the data directory to the proper location. In addition, the `data` descriptor may require hand tooling.

2. Conversion does not occur against the new DII directory structure when using ConvertSeg. For instance, if the `progs` directory exists instead of the `bin` directory, no error is generated and no translation occurs. Path names remain the same in the descriptor files and in the directory.

3. ConvertSeg does not create a `SegDescrip.NEW` directory as identified in the *DII COE Integration and Runtime Specification*; instead, it moves the original `SegDescrip` directory to `SegDescrip.orig` and creates a converted `SegDescrip` directory.

MakeInstall Tool

1. MakeInstall supports multi-volume tapes, but it will not divide segments across tapes.

2. MakeInstall is case sensitive. Segment names and prefixes must match exactly.

3. MakeInstall does not allow segments and their descriptors to be checked out from the repository.

4. MakeInstall does not support the `-Cd`, `-Co`, `-dio`, `-Dio`, `-ot`, `-R`, `-so`, `-So`, and `-T` command line options.

5. The `TOOLS_HOME` environment variable must be set for the device table to be displayed.

6. In order to *not* compress a segment, the `$NOCOMPRESS` option must be specified in the `Direct` descriptor entry in the `SegInfo` file. MakeInstall no longer prompts the user with a compress option; instead, the tool automatically references the `SegDescrip` entry.

7. Components of an Aggregate segment (parent and all children) must be contained within the distribution file or completely external to the distribution file.

8. The `SEG_DIRS` environment variable must be set if the `-p` parameter is not used. In addition, the `SEG_DIRS` environment variable is not supported for multiple distribution files; as a result, when specifying multiple distribution files, the files must all reside in the current working directory. For ease of use, set the `SEG_DIRS` environment variable to contain the paths where the segments exist.

TestInstall Tool

1. The keywords "`PC`" or "`Pentium`" in the `$CPU` field and "`NT`" or "`SCO`" in the `$OPSYS` field are not valid on UNIX platforms.

2. Network Information Service (NIS) and Domain Name Service (DNS) servers are not updated when processing the `Network` descriptor.

3. The `$NETMASK` keyword in the `Network` descriptor is not processed.

4. The `$HOME_DIR` keyword in the `Requires` descriptor is not processed.

5. TestInstall does not support distributions. It supports segment test installation only.

TestRemove Tool

1. TestRemove does not remove services, groups, or password file entries and does not perform an unmount.

2. TestRemove does not support distributions. It supports segment test removal only.

TimeStamp Tool

1. If a valid version line in the version file cannot be identified and no command line parameters are given, the TimeStamp tool will return an error.

VerifySeg Tool

1. VerifySeg does not check the syntax of version numbers listed in the `Compat` Segment Descriptor file.

2. VerifySeg performs keyword parsing (searches for keywords and their associated fields), but does not perform validity (bounds or data field) checking in the `Conflicts` or `VERSION` descriptor file.

3. VerifySeg does not perform checking in the `FilesList` Segment Descriptor file.

4. VerifySeg performs validity checking (bounds or data field) on keywords, but not on data associated with keywords (for example, disk size in the `Hardware` Segment Descriptor file).

5. VerifySeg does not check keyword combination or keyword and segment type constraints in the `Hardware` Segment Descriptor file.

VerUpdate Tool

1. If a valid version line in the version file cannot be identified and no command line parameters are given, the VerUpdate tool will return an error.

Public APIs

Please refer to the *DII COE API Reference Guide (Solaris 2.5.1)* and the *DII COE Programmer's Manual (Solaris 2.5.1)* for information on public APIs. For information on APIs specific to Helper Function, refer to the *DII COE Programmer's Manual for Helper Function (HP-UX 10.20/Solaris 2.5.1).*

This page intentionally left blank.